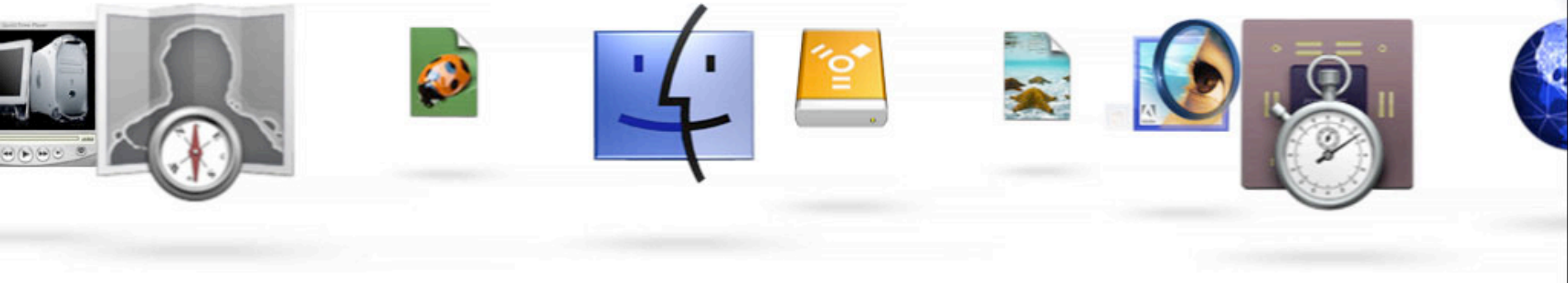




Using Interface Builder

Session 904





Using Interface Builder

Henri Lamiroux
Engineering Manager

Agenda

- What is Interface Builder (or IB for short)?
- Developing an application with Interface Builder
 - Laying out the user interface
 - Connecting the user interface to code
 - Loading the user interface
- Tips and Tricks
- CodeWarrior integration



Agenda

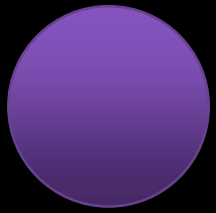
- **What is Interface Builder?**
- Developing an application with Interface Builder
 - Laying out the user interface
 - Connecting the user interface to code
 - Loading the user interface
- Tips and Tricks
- CodeWarrior integration



What Is Interface Builder?

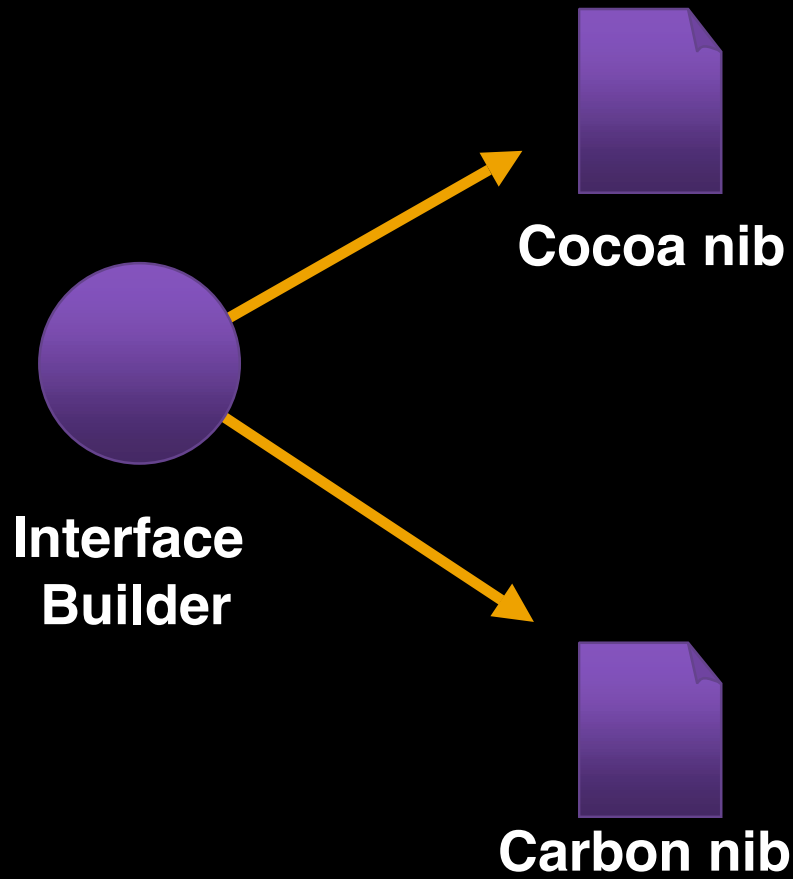
- Apple's User Interface Builder Tool
- Create Aqua-compliant User Interfaces for Cocoa and Carbon
- Recently added: AppleScript

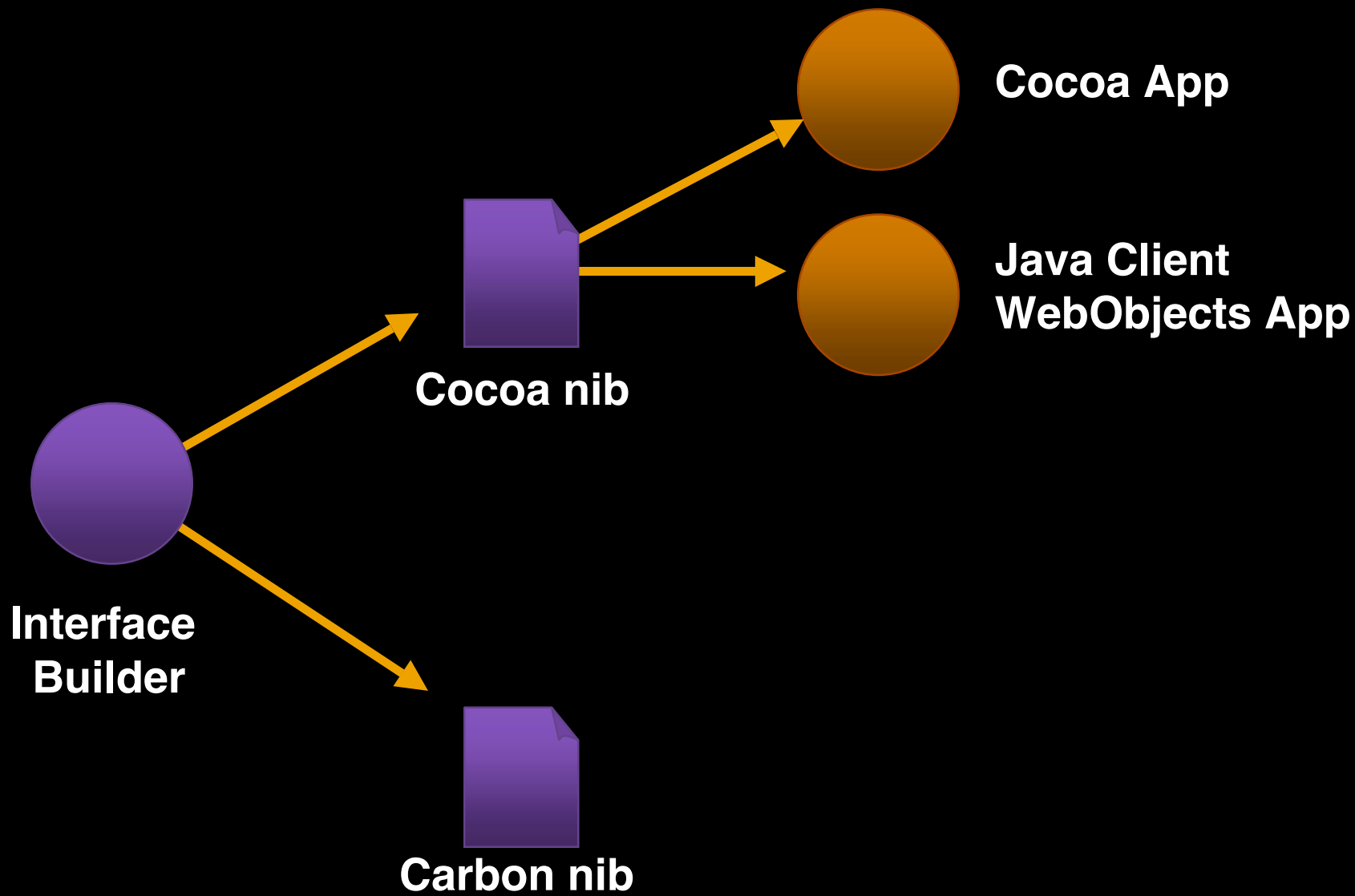


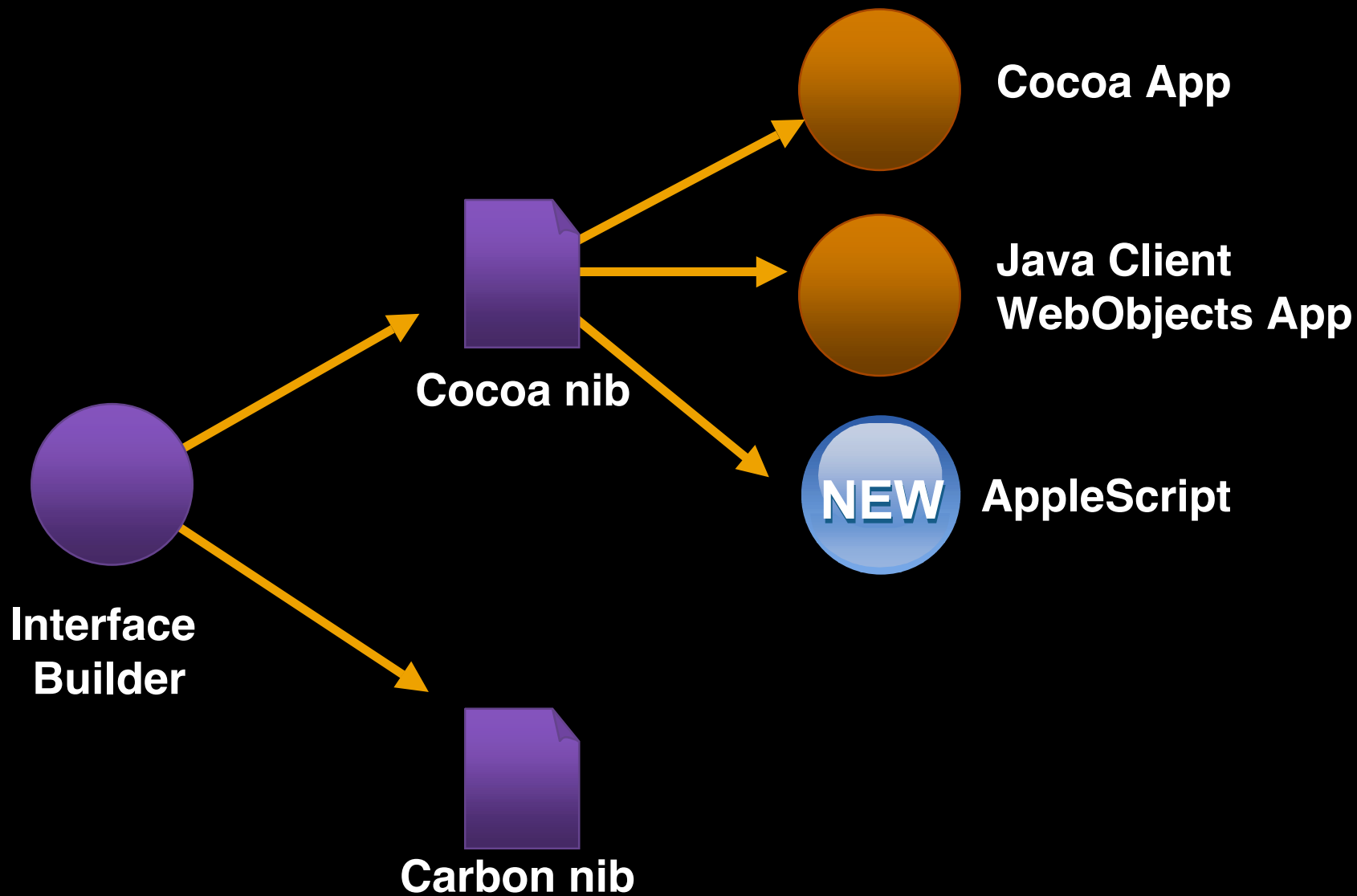


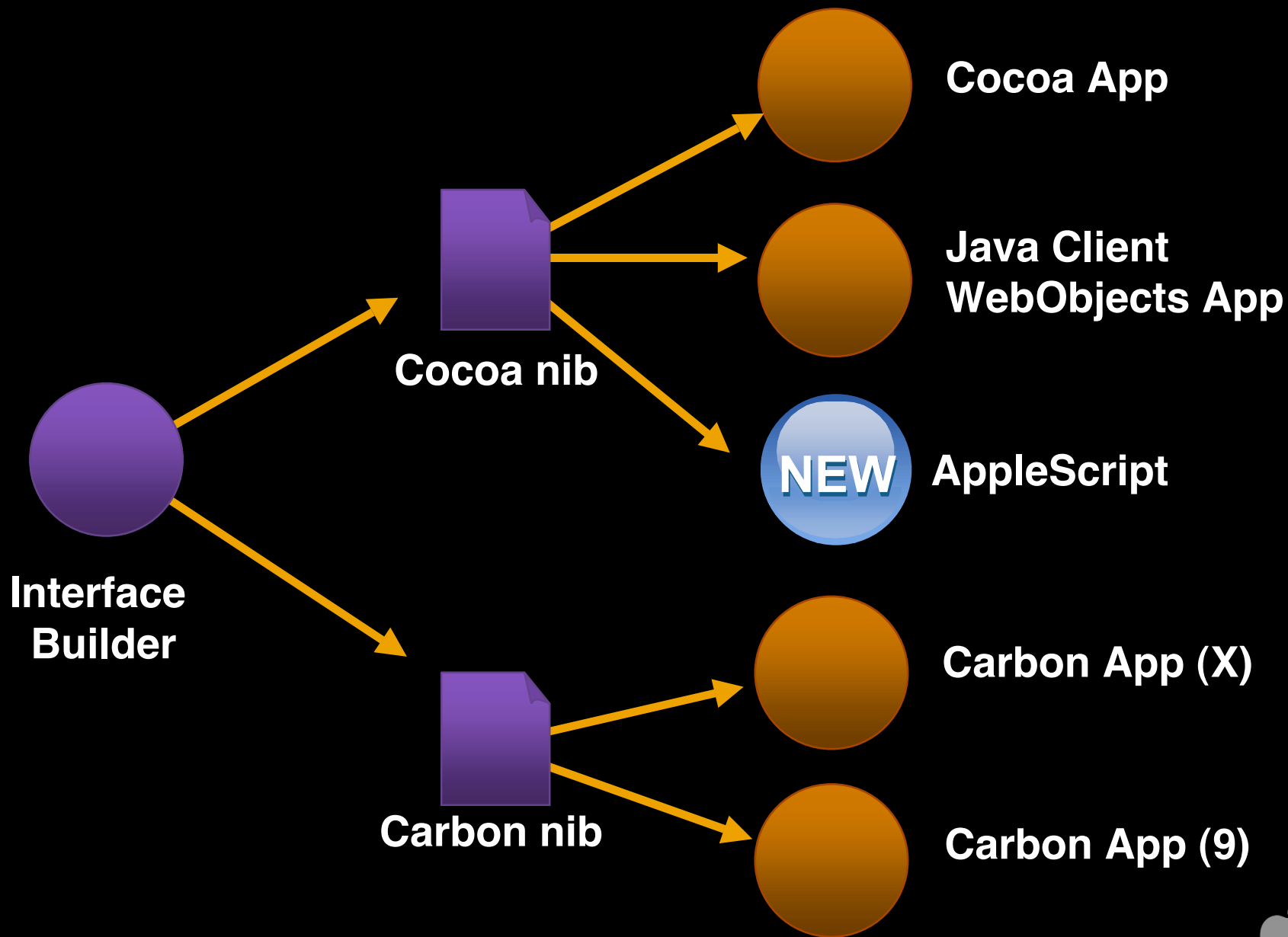
Interface Builder











Agenda

- What is Interface Builder?
- Developing an application with Interface Builder
 - **Creating the user interface**
 - Connecting the user interface to code
 - Loading the user interface
- Tips and Tricks
- CodeWarrior integration



Creating the User Interface

- Guides
- Layout Rect
- Distance Measurement
- Logical Grouping
- Locking
- Pop and Drag
- Springs and Struts (Cocoa)



Guides

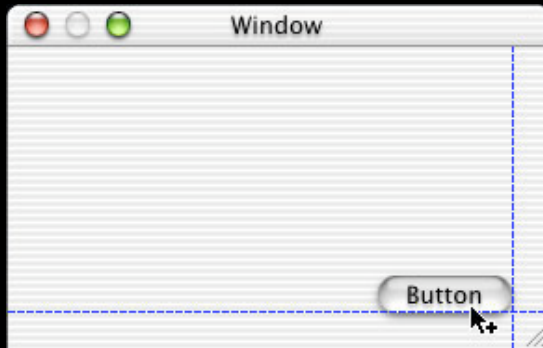
- HIG Compliance
- Much faster and intuitive way to align objects
- Two types:
 - Aqua Guides
 - User Guides



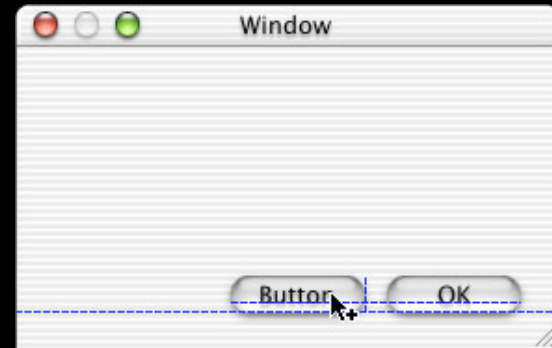
Guides

- Aqua Guides
 - Implements Aqua layout guidelines

(1)



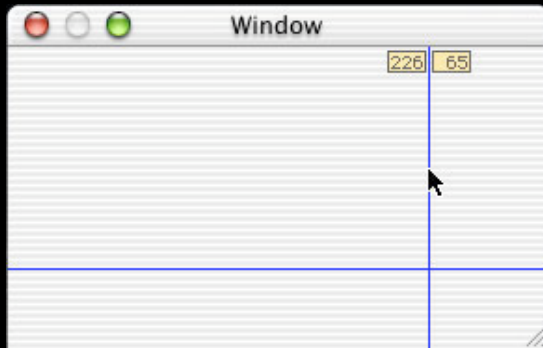
(2)



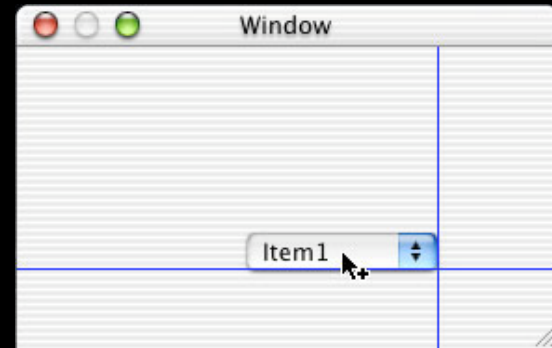
Guides

- User Guides
 - User-defined guides
 - Saved with the nib file

(1)

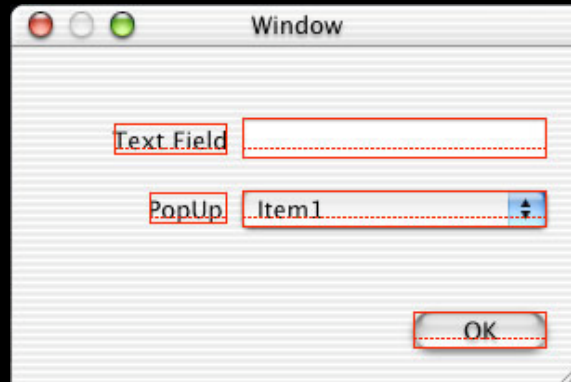


(2)



Layout Rect

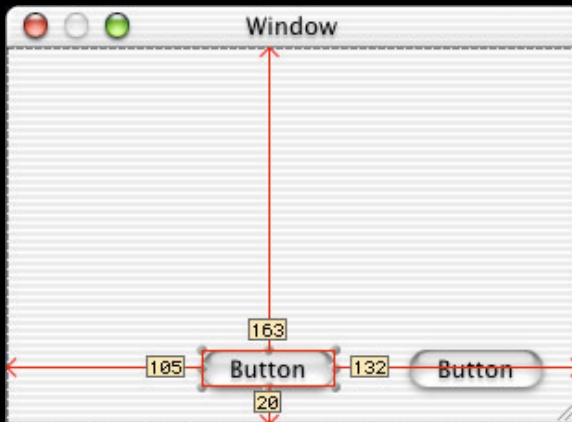
- Visual bounding box of an object (without shadow)
- Matches Aqua Guidelines
- Show/Hide with Command-L



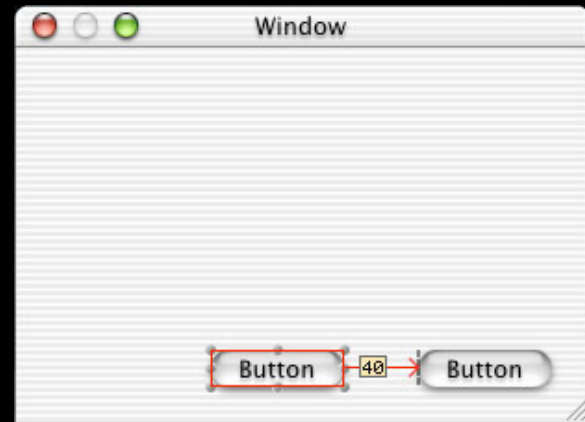
Distance Measurement

- Show distance between selected object and object under the cursor
- Triggered with Option key

(1)



(2)





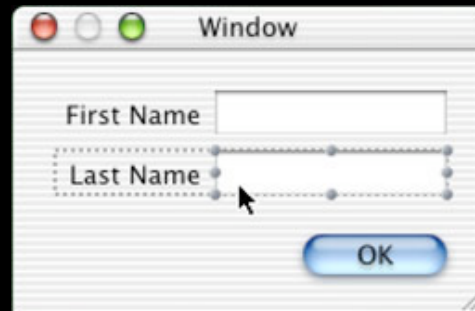
Demo

Henri Lamiroux
Engineering Manager

Logical Grouping



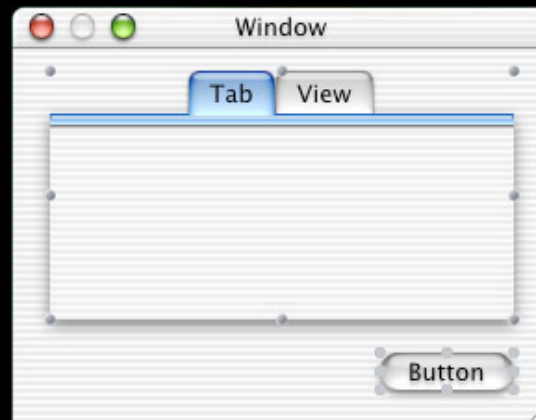
- Enables movement of multiple objects as a group without creating a container



Locking



- Position and Size
- Views(Controls)/Windows



Pop and Drag



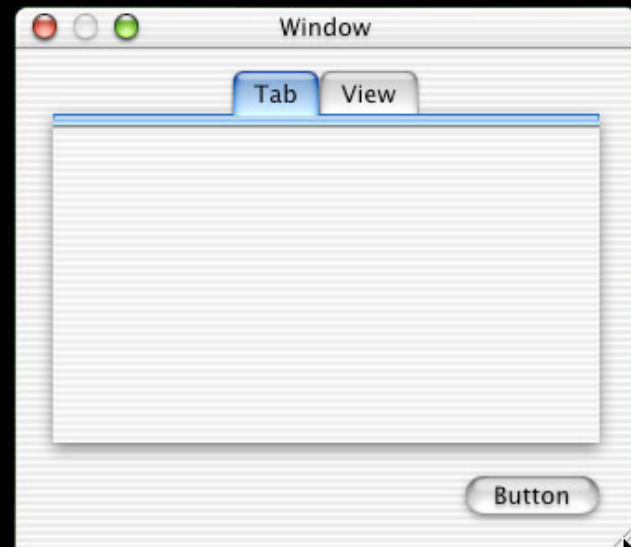
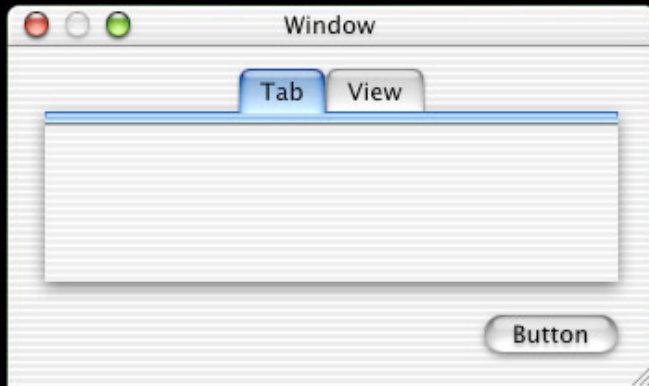
- Enables movement of controls between containers (Views and Menus)
- Maintains all connections (Cocoa)



Springs and Struts (Cocoa)



- Resize Window with Control-Key pressed





Demo

Henri Lamiroux
Engineering Manager

Agenda

- What is Interface Builder?
- Developing an application with Interface Builder
 - Creating the user interface
 - **Connecting the user interface to code**
 - Loading the user interface
- Tips and Tricks
- CodeWarrior integration



Connecting the UI to Code

- Different programming models
 - Carbon: Procedural (C APIs)
 - Cocoa: Object oriented
 - Relies on dynamic features of Objective-C



Connecting UI to Code—Carbon

- IB does not change the programming model
- Better way to layout and package the user interface
- Easy to take advantage of the Carbon Event Model



Connecting UI to Code—Carbon

- Getting a reference to UI elements
- Reacting to user actions



Connecting UI to Code—Carbon

- Getting a reference to UI elements
 - In Interface Builder assign:
 - Control ID to controls
 - Menu ID to menus
 - In your code, call the appropriate API
 - `GetControlByID(. .)` .
 - `GetMenuRef(. .)`, etc



Connecting UI to Code—Carbon

- Reacting to user actions
 - In Interface Builder assign a Command ID to:
 - Menu Items
 - Controls
 - Implement Event Handlers to process those commands





Demo

Henri Lamiroux
Engineering Manager

Connecting UI to Code—Cocoa

- Four important Cocoa/IB concepts:
 - Outlet
 - Target/Action
 - File's Owner
 - First Responder
- } Connections



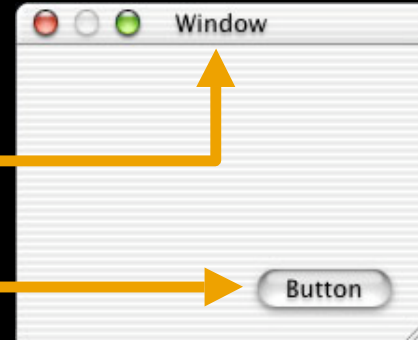
Connecting UI to Code—Cocoa

- Outlet
 - An instance variable
 - Graphically initialized by a connection in Interface Builder



Connecting UI to Code—Cocoa

```
@interface foo {  
    NSWindow* window;  
    NSButton* button;  
}  
....  
@end
```

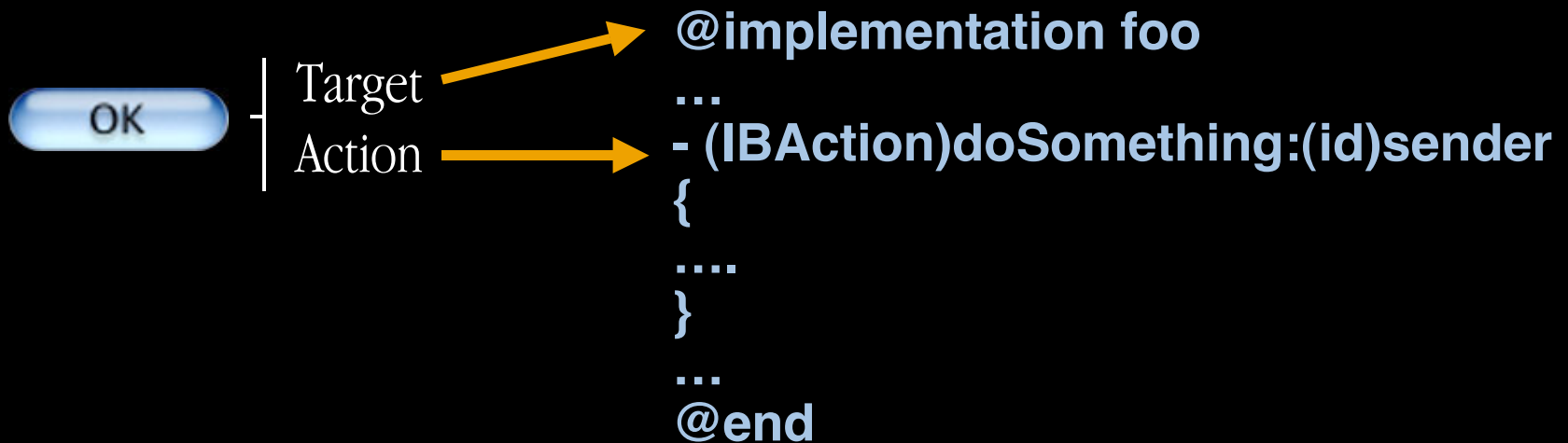


Connecting UI to Code—Cocoa

- Target
 - An object
 - Responsible for responding to a user's interaction with an interface element
- Action
 - A target method
 - Invoked by the user's interaction
- Graphically connected in Interface Builder



Connecting UI to Code—Cocoa



Connecting UI to Code—Cocoa

- Connections are saved in the nib file
- At runtime, all connections are resolved
 - Outlets are initialized
 - Controls/menu items are ready to trigger the appropriate code



Connecting UI to Code—Cocoa



- Typed Outlets
 - Improves code clarity and correctness
 - Use 'id' for previous behavior

The screenshot shows the Xcode interface for the NSWindow class. The 'ClassName' field is set to 'NSWindow'. Below it, there are two tabs: '3 Outlets' (selected) and '13 Actions'. A table lists the outlets:

Outlet Name	Type
delegate	id
initialFirstResponder	NSResponder
toolbar	NSToolbar



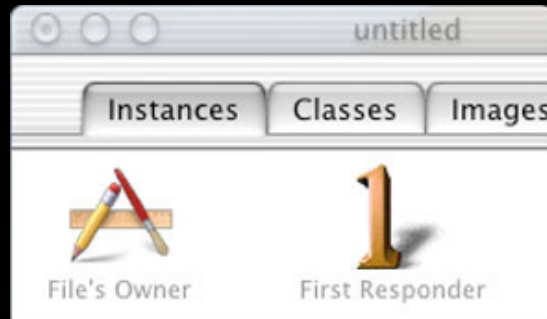


Demo

Henri Lamiroux
Engineering Manager

Connecting UI to Code—Cocoa

- File's Owner and First Responder
 - Proxy objects
 - Represent an object
 - Identity defined at runtime



Connecting UI to Code—Cocoa

- File's Owner

- Existing object, external to the nib
- Specified when loading the nib

```
[NSBundle loadNibNamed:@"myNib"owner:theOwner];
```

- Allows connections between objects defined in the nib file with objects external to the nib file



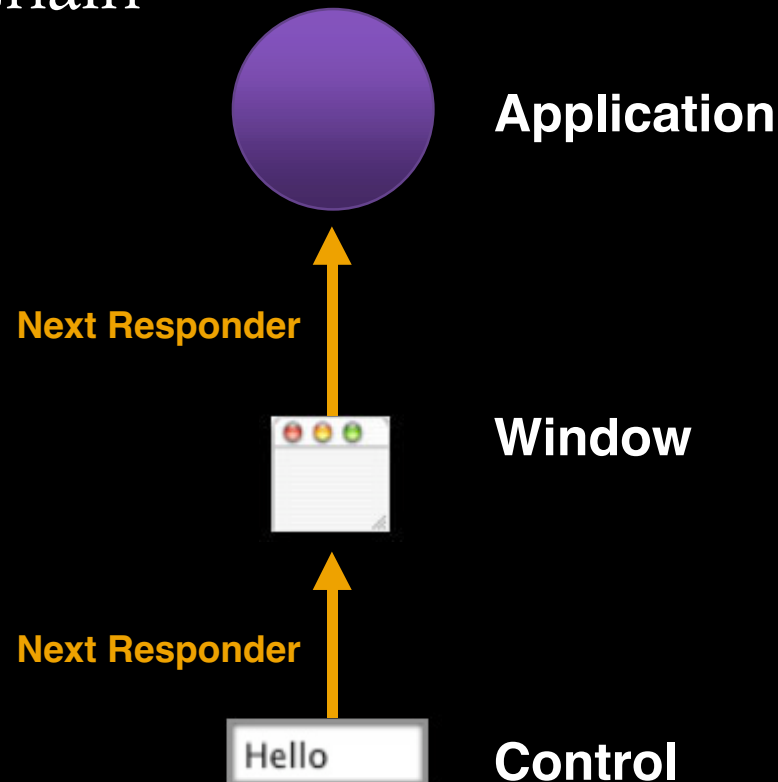
Connecting UI to Code—Cocoa

- First Responder
 - A status conferred to an object at runtime
 - First object in the responder chain to respond to a message
 - Changes based upon
 - Focus
 - Message



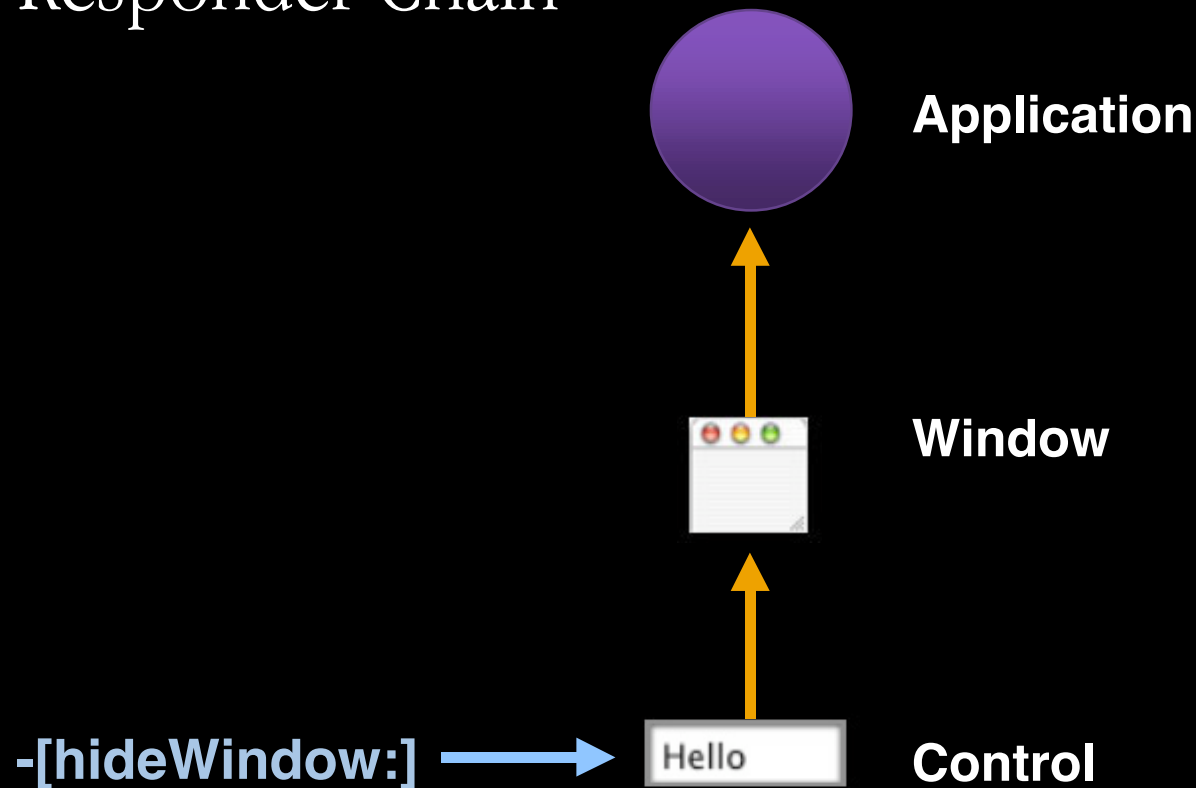
Connecting UI to Code—Cocoa

- Responder Chain



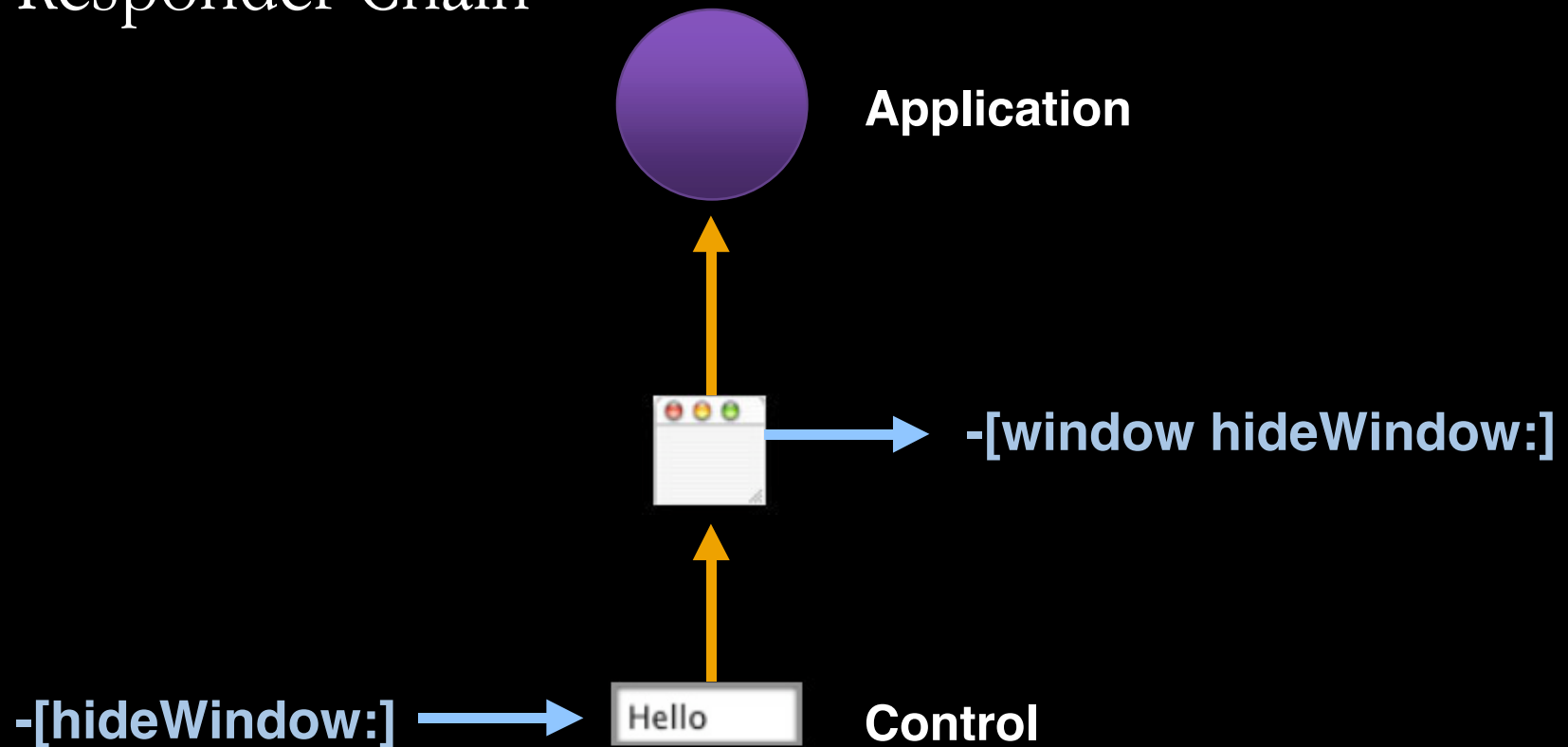
Connecting UI to Code—Cocoa

- Responder Chain



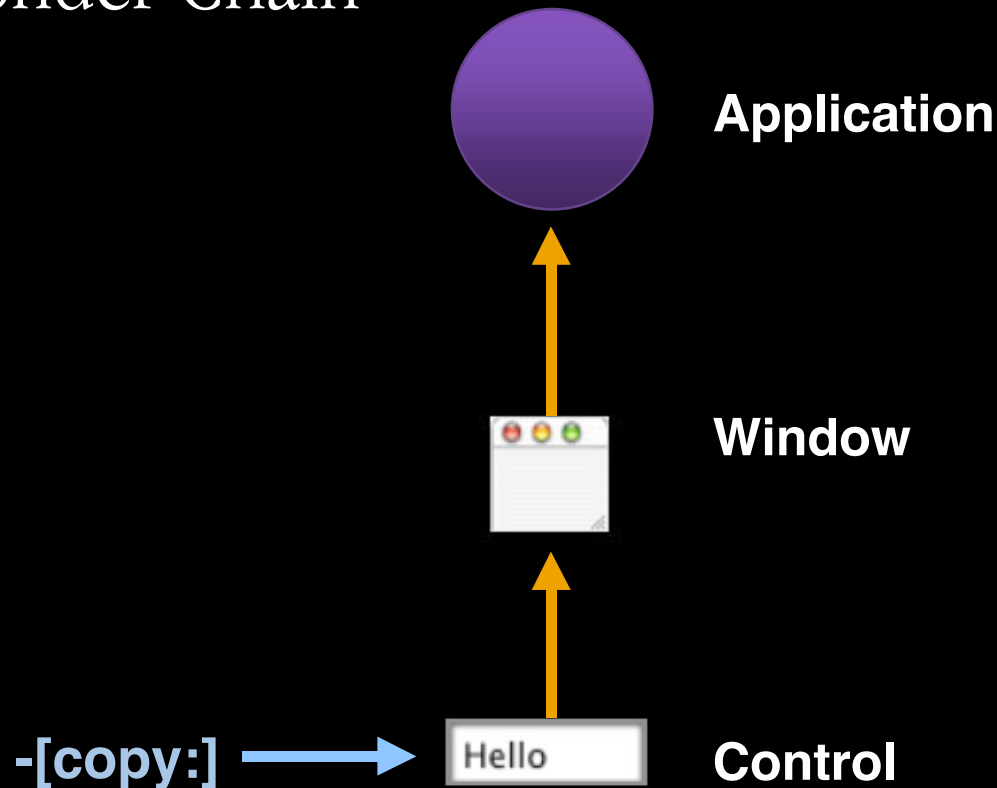
Connecting UI to Code—Cocoa

- Responder Chain



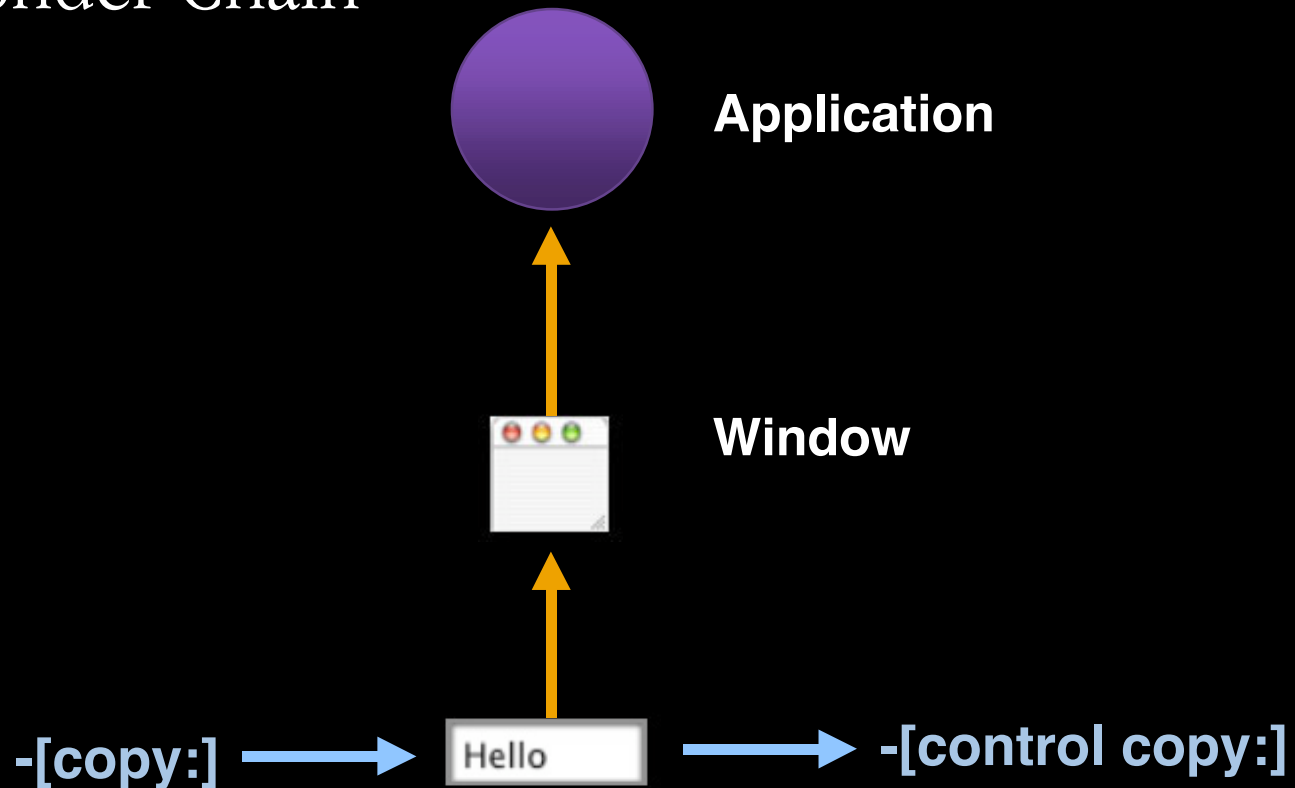
Connecting UI to Code—Cocoa

- Responder Chain



Connecting UI to Code—Cocoa

- Responder Chain



Connecting UI to Code—Cocoa

- First Responder is useful for connecting an action:
 - To an unspecified instance of a target
 - Cut/Copy/Paste
 - Hide Window
 - Etc . . .
 - To an object in another nib
 - As long as this object is in the responder chain





Demo

Henri Lamiroux
Engineering Manager

Agenda

- What is Interface Builder?
- Developing an application with Interface Builder
 - Creating the user interface
 - Connecting the user interface to code
 - Loading the user interface
- Tips and Tricks
- CodeWarrior integration



Loading the User Interface

- Different approach between Cocoa and Carbon
 - Carbon
 - Always explicit
 - Cocoa
 - Automatic for the “main” nib
 - Other nibs loaded programmatically



Loading the UI—Carbon

- One Header: IBCarbonRuntime.h
- 6 Functions
- On Mac OS X: part of Carbon framework
- On Mac OS 9: part of CarbonLib (v 1.1 or later)



IB Carbon Runtime APIs

```
int main(int argc, char* argv[]) {  
    INibRef    nibRef;  
    WindowRef window;  
  
    CreateNibReference (CFSTR("Main"), &nibRef);  
  
    SetMenuBarFromNib (nibRef, CFSTR("MenuBar"));  
  
    CreateWindowFromNib (nibRef,  
                          CFSTR("MainWindow"),  
                          &window);  
  
    DisposeNibReference (nibRef);  
  
    ShowWindow(window);  
    RunApplicationEventLoop();  
}
```



Loading the UI—Cocoa

- One special nib file: the “main” nib
 - Automatically loaded at launch time
 - Defined in Project Builder (“Application Settings” tab)
 - Typically contains the menu bar
 - File’s Owner will be the application object



Loading the UI—Cocoa

- Other nibs should be loaded lazily
 - Using NSBundle directly
 - `-[NSBundle loadNibNamed:owner:]`
 - Using NSWindowController
 - Check “SimpleMultiWindow” example in `/Developer/Examples/InterfaceBuilder`



Agenda

- What is Interface Builder?
- Developing an application with Interface Builder
 - Creating the user interface
 - Connecting the user interface to code
 - Loading the user interface
- **Tips and Tricks**
- CodeWarrior integration

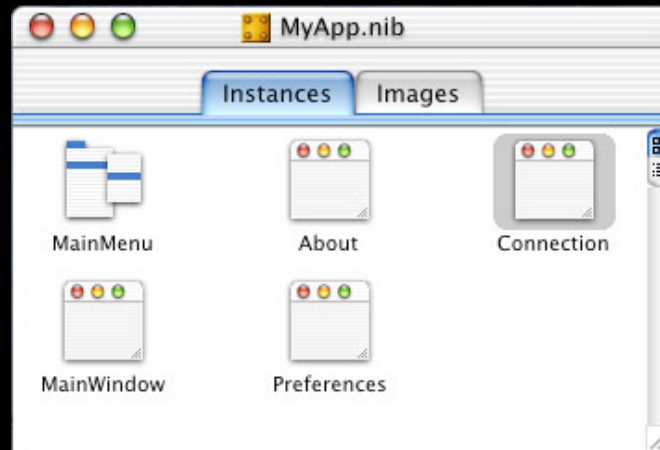


1 Breakup Your Nibs

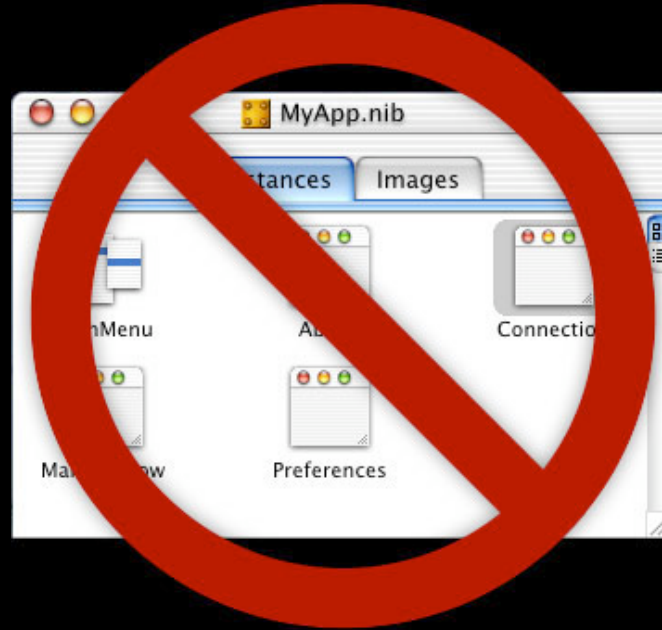
- Do not put your entire UI in one nib
 - All objects in a nib file are instantiated at once
 - Your nib files will load faster if broken up
 - Facilitates re-use



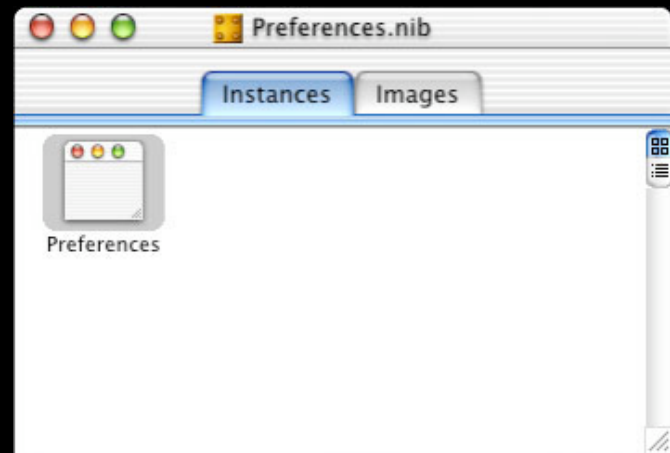
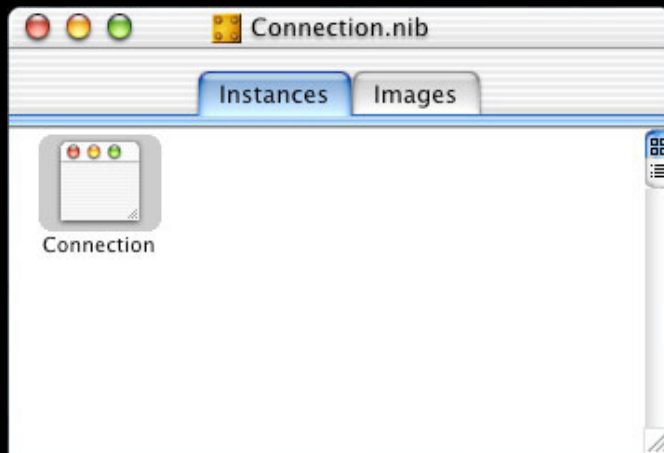
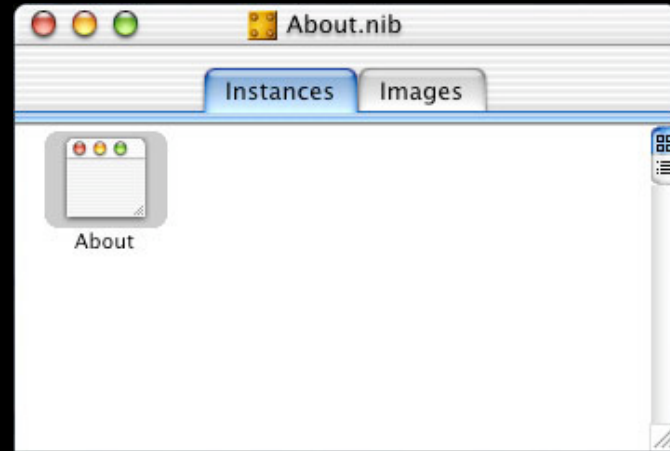
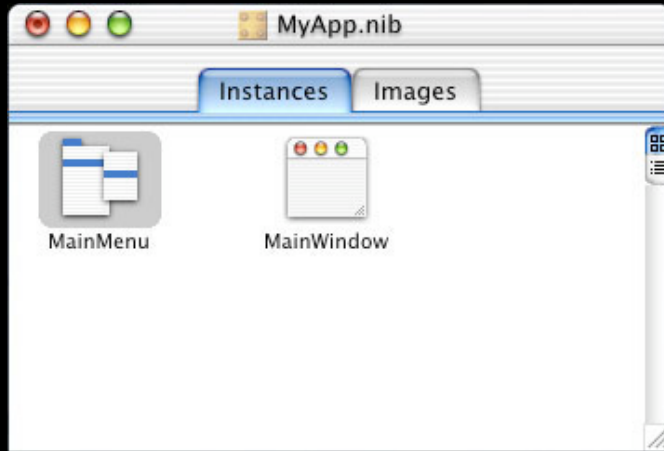
Breakup Your Nibs



Breakup Your Nibs



Breakup Your Nibs



2 Top Level Objects—Cocoa

- Objects you **instantiated** within IB
 - Windows
 - MenuBar
 - Controllers
 - . . .



Top Level Objects—Cocoa

- Be aware of their reference count
 - Default to one
- Someone must release those objects
 - Usually the responsibility of the File's Owner
 - `NSWindowController` does it for you
 - One exception: `NSWindow` has a “release on close” attribute



3 Standalone Views—Cocoa

- Allows a view to be instantiated without an enclosing window
 - Drag a Custom View to the document window
- Eliminates the need for creating a window in order to use only its content view
- Useful for:
 - Content view for NSDrawer
 - Toolbar custom items
- Be careful with reference counting



4 Initialization—Cocoa

- Post nib loading initialization
 - Implement `-[awakeFromNib]`
 - All objects have been instantiated
 - All outlets, targets/actions have been set
 - Do not rely on
 - `-[init]`
 - `-[initWithFrame:]`
 - `-[initWithCoder:]`



5 New nib Format—Cocoa

- Takes advantage of new archiving mechanism
- Provides forward and backward compatibility
- Only available in Jaguar



New nib Format—Cocoa



- nib format compatibility preferences:
 - 10.1.x only
 - Will work on all versions
 - No access to new Cocoa features
 - Jaguar and later only
 - Will work on Jaguar and later version only
 - Both
 - Both format are saved (bigger nib)



5 Images—Carbon

- Images need to be in your Project Builder project in the same target as the nib file
- Must be in compiled resource format (.rsrc extension)
- Supports
 - ICON
 - Icns
 - Cicc
 - PICT



6 nibTool

- Lets you manipulate the contents of a nib file from the command line
- Works with both Cocoa and Carbon nibs
- Look at nibtool manpage for all options
- Used by File Merge to diff nib files
- Use nibtool to localize nibs



Localization With nibtool



English nib



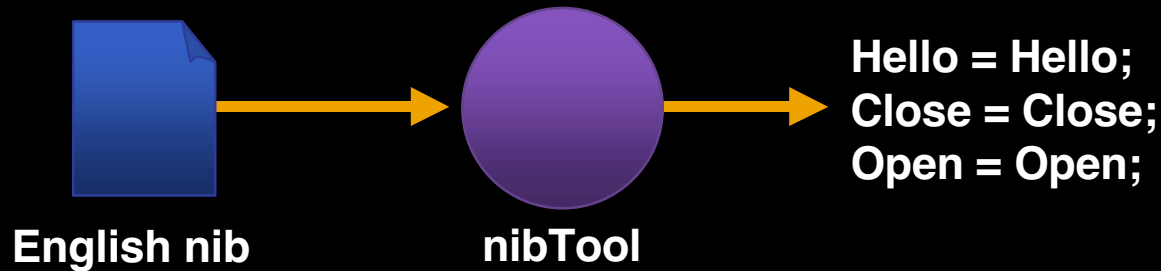
?



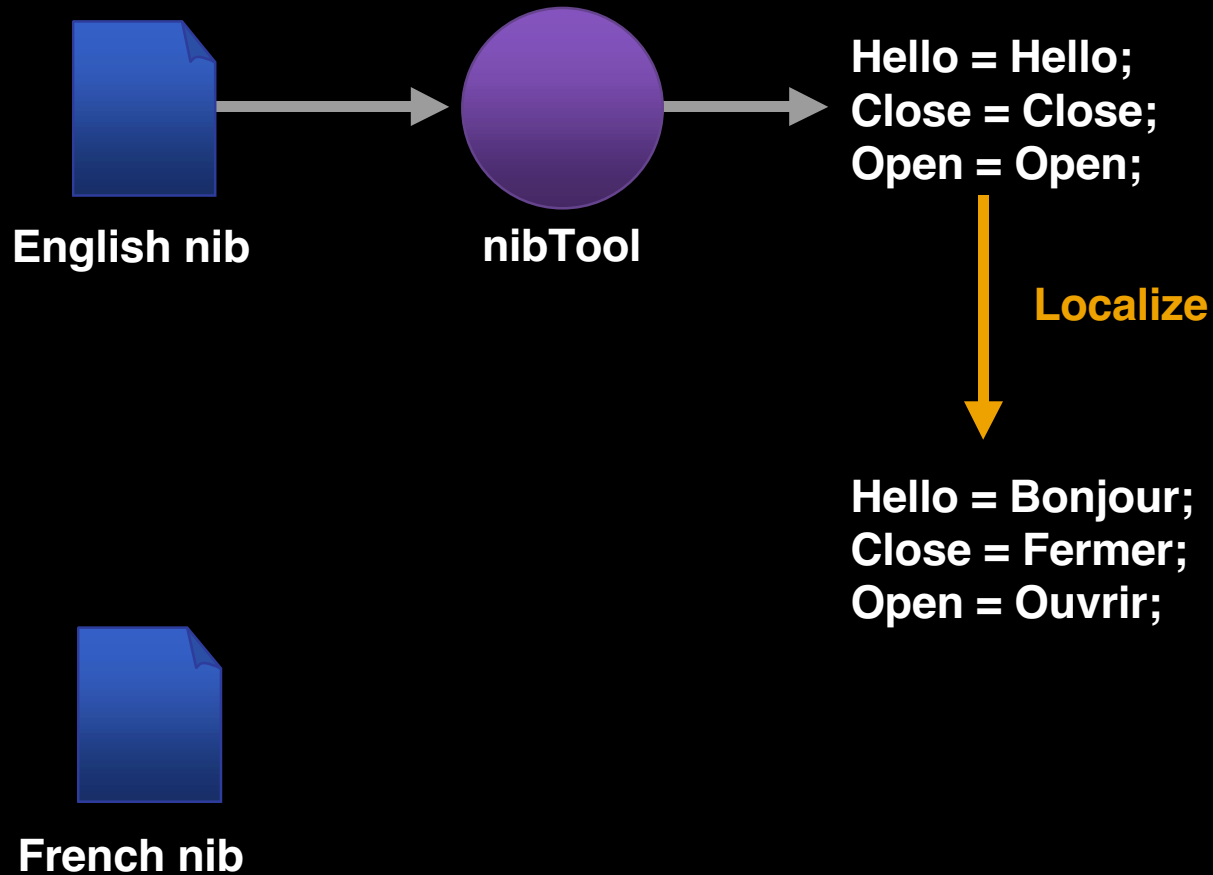
French nib



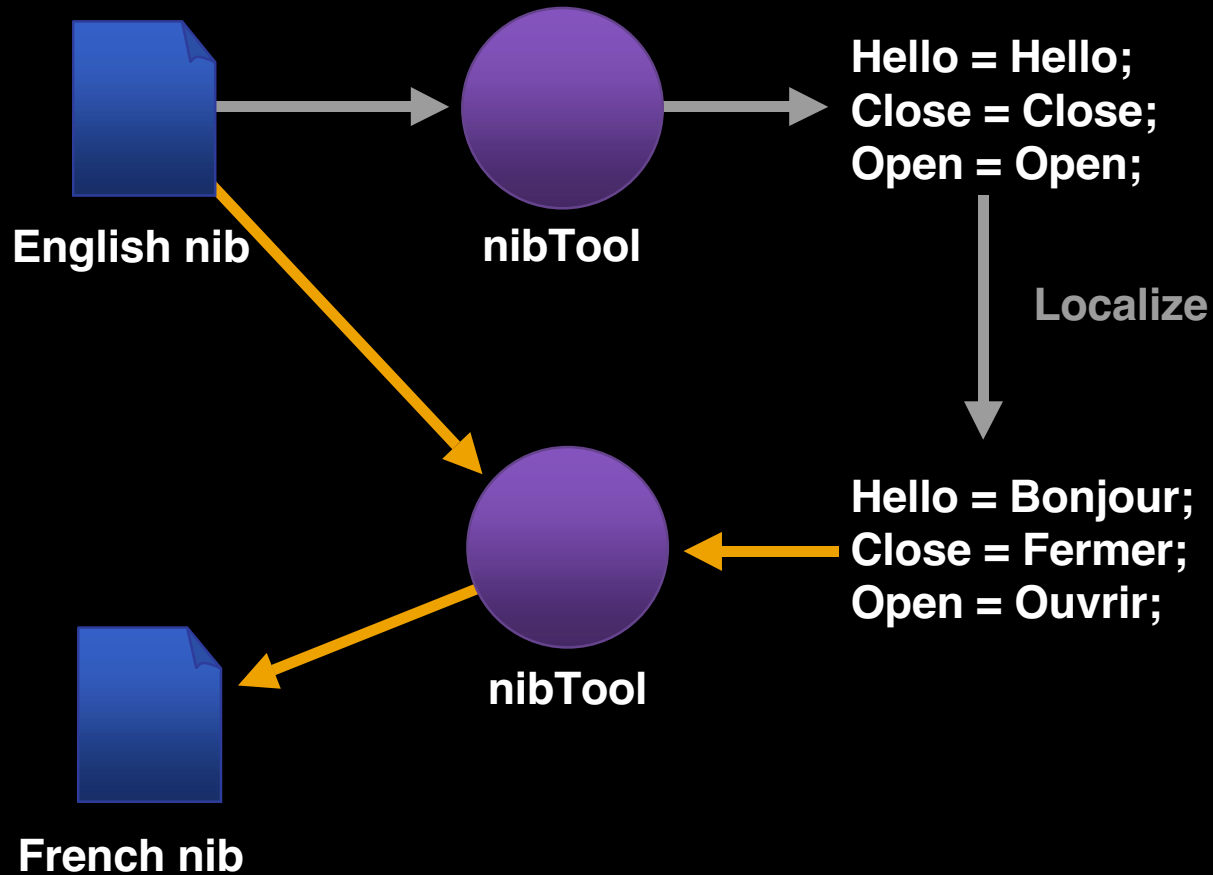
Localization With nibtool



Localization With nibtool



Localization With nibtool



Agenda

- What is Interface Builder?
- Developing an application with Interface Builder
 - Creating the user interface
 - Connecting the user interface to code
 - Loading the user interface
- Tips and Tricks
- CodeWarrior integration





CodeWarrior Integration

Isaac Wankerl
Metrowerks

CodeWarrior and Interface Builder

- CW and IB: Two great tastes that taste great together!
 - Use CodeWarrior to build Cocoa or Carbon nib-based projects
 - Quickly design your user interface with Interface Builder
 - Quickly build your app with CodeWarrior





Demo

CodeWarrior and Interface Builder

Documentation

Interface Builder

- Interface Builder Help
Help menu > Interface Builder Help
- Interface Builder Tutorials
/Developer/Documentation/DeveloperTools/InterfaceBuilder
/Developer/Documentation/Cocoa/JavaTutorial
/Developer/Documentation/Cocoa/ObjCTutorial
- Interface Builder Website at
<http://developer.apple.com/tools/interfacebuilder>



Roadmap

905 Apple Performance Tools:

See the tools to optimize your application

Hall 2

Thurs., 5:00pm

906 Developing for Performance:

Understand system performance concepts

Hall 2

Fri., 9:00am

305 Cocoa Drawing:

Learn how to work with graphics easily

Hall 2

Fri., 10:30am

FF015 Development Tools:

Make your thoughts known

Room J1

Fri., 3:30pm



Who to Contact

John Geleynse

User Experience Evangelist

Apple Worldwide Developer Relations

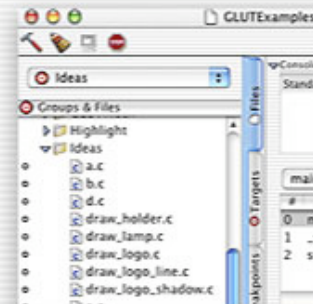
geleynse@apple.com

<http://developer.apple.com/wwdc2002/urls.html>





Q&A



John Geeynse
User Experience Evangelist
geeynse@apple.com

<http://developer.apple.com/wwdc2002/urls.html>

 **WWDC2002**

 **WWDC2002**

 **WWDC2002**